

Министерство образования и науки Донецкой Народной Республики
Государственное профессиональное образовательное учреждение
«Донецкий политехнический техникум»
Цикловая комиссия программирования и информатики

МЕТОДИЧЕСКАЯ РАЗРАБОТКА ОТКРЫТОГО ЗАНЯТИЯ
«Организация ветвлений в программах»
по дисциплине
«Основы программирования»

Специальность 09.02.03 «Программирование в компьютерных системах»

2016

Методическая разработка открытого занятия по дисциплине «Основы программирования» для образовательных учреждений СПО по специальности 09.02.03 «Программирование в компьютерных системах», 2016

Подготовила Мамедова Н.Д. – преподаватель квалификационной категории «Специалист высшей категории» цикловой комиссии информатики и программирования ГПОУ «Донецкий политехнический техникум».

Методическая разработка составлена в соответствии с ОПП на основании нормативной программы дисциплины подготовки младшего специалиста по специальности 09.02.03 «Программирование в компьютерных системах»

Для преподавателей ГПОУ «Донецкий политехнический техникум»

Рецензенты: к.т.н., преподаватель кафедры автоматизированных систем управления ДонНТУ Червинская Н.В., преподаватель высшей категории ГПОУ «Донецкий политехнический техникум» Кобцева Ю.В.

Методическая разработка рассмотрена и одобрена на заседании цикловой комиссии информатики и программирования ГПОУ «Донецкий политехнический техникум».

Протокол № 6 от 04.02.2016 г.

Специальность: 09.02.03 «Программирование в компьютерных системах»"

Дисциплина: Основы программирования

Преподаватель: Мамедова Н.Д.

ПЛАН ЗАНЯТИЯ

Тема занятия: «Организация ветвлений в программе»"

Цель занятия:

Методическая: продемонстрировать отработки методики организации и проведения семинара-практикума с использованием интерактивных технологий

Дидактическая: систематизация и обобщение знаний студентов по знанию особенностей операторов для организации ветвлений на языке C / C ++, закрепление теоретических знаний по синтаксису и особенностям использования условных операторов и оператора множественного выбора, углубление знаний по программированию алгоритмов для организации ветвлений в программе, овладение различными приемами программирования с использованием этих операторов при решении прикладных задач

Развивающая: развитие у студентов алгоритмического и логического мышления, умения анализировать, сравнивать, развитие умения поиска закономерностей, поиска оптимальных решений

Воспитательная: формирования культуры умственного труда, формирования эмоционально-ценностного отношения к практической деятельности, воспитания творческой активности.

Вид занятия: семинар по углублению, обобщения и систематизации знаний

Форма и методы проведения занятия: решение проблемных задач, эвристическая беседа

Междисциплинарные связи:

Обеспечивающие основы информатики, математический анализ, геометрия, алгебра

Обеспечиваемые алгоритмы и структуры данных, объектно-ориентированное программирование

Методическое обеспечение:

методическая разработка занятия, презентация к занятию, опорный конспект, раздаточный материал, компьютерная программа для создания презентаций MS Power Point, тестовая оболочка для конструирования тестов, среда программирования (MS Visual C ++, Turbo C ++ или др.)

Технические средства обучения: интерактивная доска, мультимедийный проектор

Литература:

Обязательная:

1. Стивен Прата «Язык программирования C ++. Лекции и упражнения ». Учебник.- СПб: ООО «ДиаСофтЮП», 2003.
2. Шпак З.Я. Программирование на языке C.-Львов: Ориана-Нова, 2006.
3. И.О.Хвищун. Программирование и математическое моделирование: Учебник. - М .: Издательский Дом «Ин Юрс», 2007.

Дополнительная:

1. Е.Л.Романов. Практикум по программированию на C ++: Уч. Пособие. СПб: БХВ-Петербург; Новосибирск: Изд-во НГТУ, 2004.
2. Л.М.Климова C ++. Практическое программирование. Решение типовых задач. - Кудиц-образ, 2001.
3. Алексеев Е.Р. MS Visual C ++ и Turbo C ++ Explorer. - М .: НТ Пресс, 2007.
4. Мультимедийный обучающий курс по изучению языка C ++ Компьютерной академии «Шаг»

Введение

Темой семинара-практикума по дисциплине «Основы программирования» является «Организация ветвлений в программе». Название темы реализуется в разделе дисциплины «Введение в C ++. Операторы языка C / C ++ », который изучается студентами 2 курса. В основу этого раздела положено изучение основных элементов языка C, средств языка C для программирования основных алгоритмических конструкций.

Комплекс знаний и практических навыков, добытых при изучении этой темы, нужен для формирования ключевых профессиональных компетенций современного квалифицированного специалиста по специальности 09.02.03 «Программирование в компьютерных системах».

При проведении занятия широко применяются мультимедийные средства обучения: интерактивная тестовая оболочка, демонстрация презентации с материалами к занятию.

Дидактической целью занятия является систематизация и обобщение знаний студентов по указанной теме, закрепление и углубление теоретических знаний, овладение различными приемами программирования при решении прикладных задач.

Воспитательной целью является формирование культуры умственного труда, формирования эмоционально-ценностного отношения к практической деятельности, воспитания творческой активности.

Особенностью занятия является то, что используются активные формы обучения, реализуется проблемный метод обучения и обеспечиваются междисциплинарные связи.

Полученные знания, базовые приемы программирования алгоритмов со сложной логикой, студенты смогут применять в дальнейшей практике, способствовать междисциплинарные интеграции знаний и умений, развития интеллектуальных умений и творческих способностей студентов.

ХОД ЗАНЯТИЯ:

1. Организационный момент

1 мин.

Проверка присутствия студентов готовности студентов и аудитории к занятию

2. Ознакомление с темой, целью и планом занятия

1 мин.

3. Мотивация обучения

2 мин.

Компьютеры используются не только для хранения данных. Они анализируют, объединяют, реорганизуют, изымают, меняют, экстраполируют, синтезируют их. Компьютер в отличие от человека, способен безошибочно осуществить многократное выполнение (цикл) одного оператора или серии операторов. В этом заключается одна из преимуществ компьютера как инструмента для облегчения умственного труда. Чтобы делать все эти преобразования, программам необходимы инструментальные средства, которые позволяют выполнять выбор и повторения этих действий. Циклы и ветвления - основная часть практически всех вычислительных процессов. Отсюда важность вопроса построения таких программ, которые оптимизированы по использованию памяти и машинного времени, верного формирования и использования величин и т.д. Сегодня мы рассмотрим различные приемы алгоритмизации и программирования с применением таких управляющих конструкций языка С, которые являются средствами для осуществления программной реализации алгоритмов с разветвляющейся структурой, сложной логикой. Рациональное использование управляющих конструкций языка является основой надежного и экономического программирования. Сегодня на занятии будут рассмотрены примеры, иллюстрируют управляющие операторы и приемы их применения при создании приложений из курса математических дисциплин.

4. Актуализация опорных знаний

20 мин.

4.1 Блиц-опрос (фронтальная работа):

3 мин

1) На какие группы делятся операторы языка С (по характеру действий, по конструкции)?

2) Каковы назначения операторов каждой классификационной группы?

3) В каких случаях следует применять условный оператор if?

4) Когда целесообразнее использовать оператор выбора switch?

5) Какие операции используются для записи условий?

Какие из операций отношения или логические имеют более высокий приоритет?

6) Каково назначение операторов break и continue?

7) Какие типы операндов и результатов операций отношений?

8) Как обозначаются и выполняются логические операции?

9) Что выполняет оператор return?

4.2 Тестовая проверка знаний студентов по теме «Ветвления» с взаимопроверкой: 10 мин

Самостоятельная работа за ПК с последующей коллективной проверкой правильности прохождения теста с применением интерактивной тестовой оболочки (см.приложение 2).

4.3 Выполнение упражнений по теме (самопроверка) 6 мин

5. Комментарий к ответам студентов 1 мин.

Комментарий ответов студентов, акцент внимания на недостатки ответов

6. Решение задач профессионального направления 40 мин.

Сопровождение демонстрацией презентации

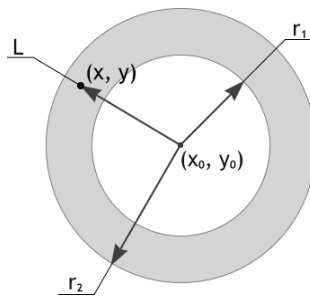
6.1 Решение геометрических задач

Задача 6.1.1.

Составить программу, которая определяет, является ли треугольник со сторонами a, b, c равносторонним, равнобедренным.

Задача 6.1.2

На плоскости нарисовано кольцо с центром в точке (x_0, y_0) . И радиусами границ $r_1 < r_2$. Кроме того, на этой же плоскости дана точка с координатами (x, y) .



Необходимо определить, принадлежит ли эта точка кольцу. Если логическое выражение $r_1 < L < r_2$ верно, то точка принадлежит кольцу. Если же хотя бы одно из этих условий не выполнено, то точка не принадлежит кольцу. Программа, реализующая это решение, приведена ниже:

```
// Точка и кольцо
#include <iostream.h>
#include <math.h>

void main() {
    // Объявление переменных
    int x0, y0, r1, r2, x, y;
    float L;

    // Пусть пользователь введет данные
```

```

    cout<<"Input coordinates of circle's center (X0,
Y0): ";
    cin>>x0>>y0;
    cout<<"Input circle radiuses R1 and R2: ";
    cin>>r1>>r2;
    cout<<"Input point coordinates (X, Y): ";
    cin>>x>>y;
    L = sqrt(pow(x - x0, 2) + pow(y - y0, 2));
    if ((r1 < L) && (L < r2 )) {
        cout<<"This point is situated inside the
circle.\n";
    } else {
        cout<<"This point is not situated inside the
circle.\n";
    }
}

```

Прокомментировать код программы.

Задача 6.1.3 . Решение квадратного уравнения

Имеется алгоритм поиска корней квадратного уравнения вида $ax^2 + bx + c = 0$..Составьте программу по готовому алгоритму

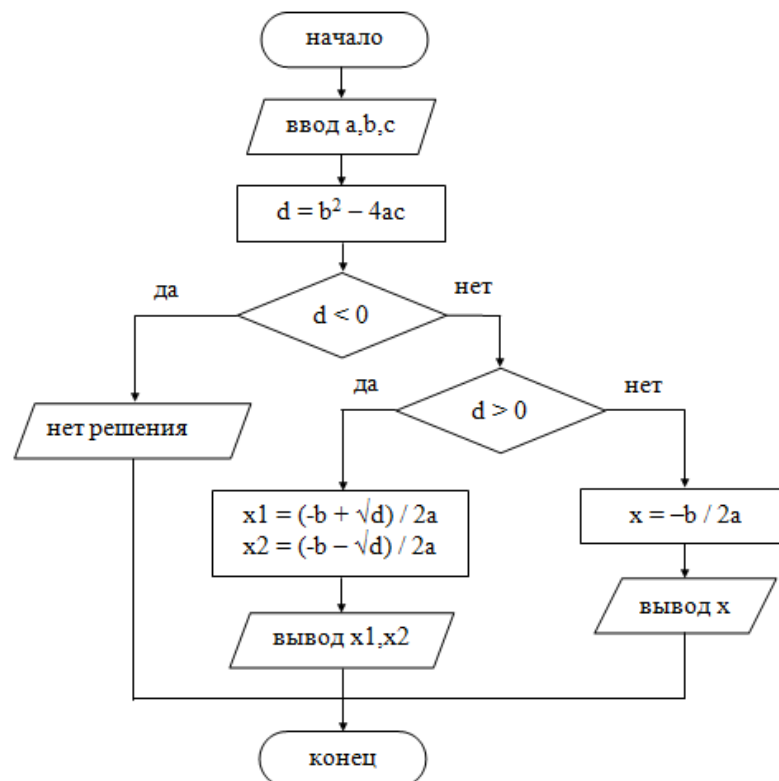


Рисунок 1 – Блок-схема к задаче 6.1.3

```
#include <iostream.h>
```



```

#include <math.h>
int main()
{double a, b, c, d, x1, x2;
cin>> a>>b>>c;
d=b*b-4*a*c;
if(d<0) cout<<"\n корней нет ;
else if(d>0) {x1=(-b+sqrt(d))/(2*a);
x2=(-b-sqrt(d))/(2*a);
cout<<"\n x1 = "<< x1;
cout<<"\n x2= "<< x2;}
else { x1=-b/(2*a);
cout<<"\n x1 = "<< x1;
}
return 0;
}

```

6.2 Элементы алгоритмизации вычислительных задач

К вычислительных задач будем относить задачи, сформулированные на математическом языке и связанные с реализацией на компьютере каких-либо математических методов решения.

Одна из характерных особенностей вычислительных задач состоит в том, что на единицу вводимой и выводимой приходится большое количество арифметических операций.

Вторая характерная особенность вычислительных задач состоит в том, что в этих задачах используется сравнительно простая структура организации данных.

Важнейшим требованием к методам и средствам решения вычислительных задач является обеспечение определенной точности вычислений. При решении задач на ЭВМ в точности понимается степень приближения результата, выданного машиной, к правильному результату.

Решения задачи, выданные машиной, могут отличаться от правильных вследствие различных причин. Основными из них являются ошибки и опечатки. Ошибки могут возникать в постановке задачи, в алгоритме и программе, а также в результате неправильной работы самой машины. Погрешности возникают из-за приближенный характер вычислительных методов, неточности исходных данных, округления при выполнении арифметических операций и др.

Вычислительные задачи решаются методами прямого вычисления и методами последовательного приближения.

Отличительная особенность методов прямого вычисления заключается в том, что ход и правильность вычислительного процесса полностью predeterminedены. Прямое вычисления - это, например, решения системы линейных алгебраических уравнений методом исключения неизвестных, вычисления площади круга с заданным радиусом, вычисления объема конуса по заданной высоте и радиусом основания и др. Отметим, что методы прямого вычисления имеют высокую точность вычисления. Однако абсолютно точное решение можно получить только для некоторых целочисленных задач.

Задача 6.2.1 Решение кубического уравнения. Демонстрация примера

Составить программу для решения кубического уравнения $ax^3 + bx^2 + cx + d = 0$

Кубическое уравнение имеет вид: $ax^3 + bx^2 + cx + d = 0$ (3.1)

После деления на a уравнение(3.1) принимает канонический вид:

$x^3 + rx^2 + sx + t = 0$, где $r = \frac{b}{a}$, $s = \frac{c}{a}$, $t = \frac{d}{a}$. В уравнении(3.2) сделаем замену $x = y - \frac{r}{3}$ и получим приведенное уравнение (3.3):

$$y^3 + py + q = 0, \text{ где } p = \frac{3s-r^2}{3}, q = \frac{2r^3}{27} - \frac{rs}{3} + t.$$

Число действительных корней приведенного уравнения(3.3) зависит от знака дискриминанта $D = \left(\frac{p}{3}\right)^3 + \left(\frac{q}{2}\right)^2$ (табл. 3.1).

Таблица 3.1 Количество корней кубического уравнения

<i>Дискримина нт</i>	<i>Кол-во действительных корней</i>	<i>Кол-во комплексных корней</i>
$D \geq 0$	1	2
$D < 0$	3	-

Корни приведенного уравнения могут быть рассчитаны по формулам Кардано (3.4):

$$y_1 = u + v$$

$$y_2 = -\frac{u+v}{2} + \frac{u-v}{2}i\sqrt{3}$$

$$y_2 = -\frac{u+v}{2} - \frac{u-v}{2}i\sqrt{3}$$

где

$$u = \sqrt[3]{-\frac{q}{2} + \sqrt{D}},$$

$$v = \sqrt[3]{-\frac{q}{2} - \sqrt{D}}.$$

При отрицательном дискриминанте уравнение (3.1) имеет три действительных корня, но они будут вычисляться через вспомогательные комплексные величины. Чтобы избавиться от этого, можно воспользоваться следующими формулами(3.5):

$$y_1 = 2\sqrt[3]{p} \cos \frac{\varphi}{3}$$

$$y_2 = 2\sqrt[3]{p} \cos(\frac{\varphi}{3} + \frac{2\pi}{3})$$

$$y_2 = 2\sqrt[3]{p} \cos(\frac{\varphi}{3} + \frac{4\pi}{3}),$$

Где

$$p = \sqrt{\frac{-p^3}{27}}$$

$$\cos \varphi = -\frac{q}{2p}(3.5)$$

Таким образом, при положительном дискриминанте кубического уравнения (3.3) расчет корней будем вести по формулам(3.4), а при отрицательном – по формулам(3.5). После расчета корней приведенного уравнения(3.3) по формулам(3.4) или (3.5) необходимо по формулам $x_k = y_k - \frac{r}{3}k = 1,2,3$ перейти к корням заданного кубического уравнения(3.1)

Описание блок-схемы.

В блоке 1 вводятся коэффициенты кубического уравнения, в блоках 2-3 рассчитываются коэффициенты канонического и приведенного уравнений.

Блок 4 предназначен для вычисления дискриминанта. В блоке 5 проверяется знак дискриминанта кубического уравнения. Если он отрицателен, то корни вычисляются по формулам (3.5) – блоки 6-7. При положительном значении дискриминанта расчет идет по формула (3.4) – блок 9. Блоки 8 и 10 предназначены для вывода результатов на экран. Программа решения кубического уравнения на C++ с комментариями приведена ниже.

```
#include <iostream.h>
#include <math.h>
#define pi 3.14159 // Определение константы
int main()
{double a, b, c, d, D, r, s, t, p, q, ro, fi, x1,
x2, x3, u, v, h, g;
// Ввод коэффициентов кубического уравнения
cout<<" a= "; cin>> a;
cout<<" b= "; cin>> b;
cout<<" c= "; cin>> c;
cout<<" d= "; cin>>d;
// расчет коэффициента канонического уравнения по
формуле (3.2)
r = b/a; s = c/a; t = d/a;
//вычисление коэффициентов приведенного уравнения
по формуле (3.3)
p = (3*s-r*r)/3; q = 2*r*r*r/27-r*s/3+t;
// вычисления дискриминанта кубического уравнения.
D=(p/3) * (p/3) * (p/3)+(q/2) * (q/2);
if(D<0)
//Формулы (3.5)
{ro = sqrt((double) (-p*p*p/27));
fi = - q/(2*ro);
fi = pi/2-atan(fi/sqrt(1-fi*fi));
x1= 2*pow(ro, (double)1/3)*cos(fi/3)-r/3;
x2= 2*pow(ro, (double)1/3)*cos(fi/3+2*pi/3)-r/3;
x3= 2*pow(ro, (double)1/3)*cos(fi/3+4*pi/3)-r/3;
cout<<"\n x1="<<x1<<"\t x2 = "<< x2<<"\t x3 =
"<<x3<<"\n";
}
else
//формулы(3.4)
{
//К выражению 1/3 необходимо применить операцию
//преобразование типа (double) 1/3, иначе будет
выполнена
```

```

        // операция целочисленного деления, результат
        // которой равен 0
        if (-q/2+sqrt(D)>0) u = pow((-q/2+sqrt(D)),
(double) 1/3);
        else if (-q/2+sqrt(D)<0) u = -pow(fabs(-
q/2+sqrt(D)), (double) 1/3);
        else u =0;
        if (-q/2-sqrt(D)>0) v =pow((-q/2-sqrt(D)), (double)
1/3);
        else
        if (-q/2+sqrt(D)<0) v = -pow (fabs(-q/2-sqrt(D)),
(double) 1/3);
        else v=0;
        // Вычисление действительного корня куб уравнения.
        x1 = u+v-r/3;
        // Вычисление действительной и мнимой части
        // комплексных корней.
        h = -(u+v)/2-r/3;
        g=(u+v)/2*sqrt((double)3);
        cout<<"\n x1 = "<< x1;
        cout<<"\t x2 = "<< h<<"+i*("<<g<<")\t x3="<<"-
i*("<<g<<")\n";
    }
    return 0;
}

```

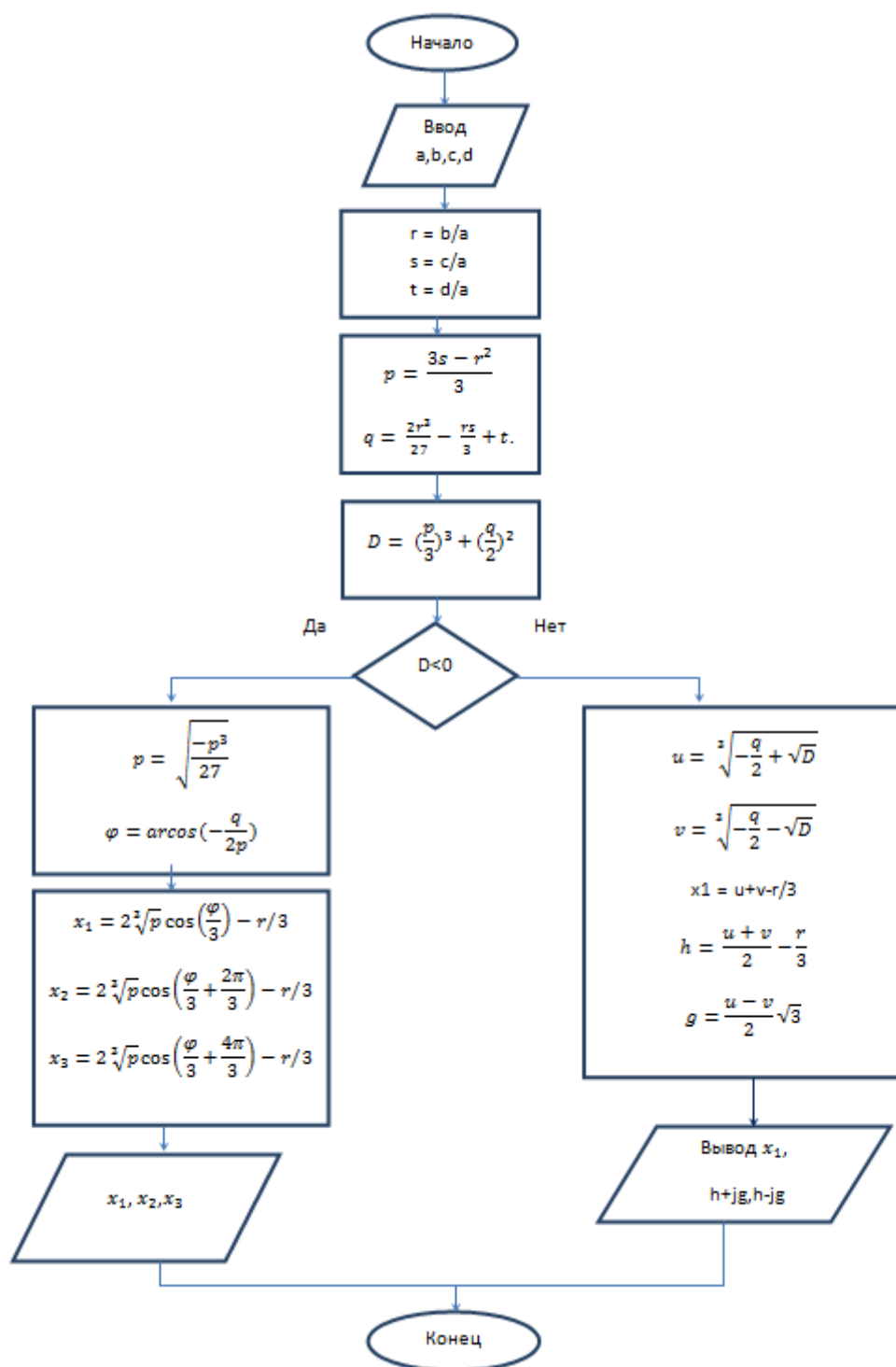


Рисунок 2 – Блок-схема к задаче 6.2.1

6.3 Задачи из курса экономических дисциплин

Задача 6.3.1 Товары на складе подлежат переоценке в соответствии с такими условиями: на товары групп 1-3 цены снижаются на 10%, на товары

группы 4 цены повышаются на 10%, на товары группы 5 цены повышаются на 15%, на товары группы 6 цены повышаются на 20%, цены на другие товары не изменяются. Определить новую цену товара по известной группе товара и старой цене.

```
#include<iostream.h>
#include<math.h>

void main()
{double zena,newzena,gr;
cout<<" Введите цену товара\n";
cin>>zena;
cout<<" Введите группу товара\n";
cin>>gr;
switch(gr)
{ case 1..3:newzena=zena-zena*0.1;break;
case 4:newzena=zena+zena*0.1;break;
case 5:newzena=zena+zena*0.15;break;
case 6:newzena=zena+zena*0.2;break;
default:cout<<"Zena          na          tovar          ne
izmenilas";newzena=zena;
}
cout<<"Staraia zena="<<zena<<"NewZena="<<newzena;
}
```

6.4 Занимательные задачи

Задача 6.4.1 Вечный календарь

Напишите программу, которая по дате определяет день недели, на который эта дата приходится. Для вычисления дня недели воспользуйтесь формулой:

$$(d + \left\lceil \frac{1}{5} (13m - 1) \right\rceil + Y + \left\lceil \frac{Y}{4} \right\rceil + \left\lceil \frac{c}{4} \right\rceil - 2c + 777) \bmod 7.$$

Здесь d — число месяца, m — номер месяца, если начинать счет с марта, как это делали в Древнем Риме (март — 1, апрель — 2, ..., февраль — 12), Y — номер года в столетии, c — количество столетий.

Квадратные скобки означают, что надо взять целую часть от значения, находящегося в скобках. Вычисленное по формуле значение определяет день недели: 1 — понедельник, 2 — вторник, ..., 6 — суббота, 0 — воскресенье.

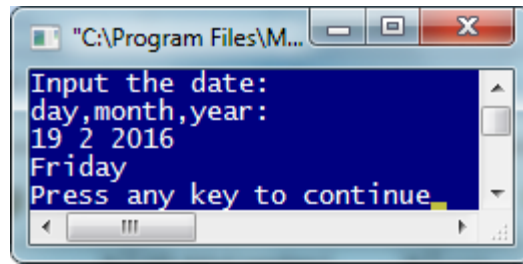


Рисунок 1 – Результат работы программы к задаче 6.4.1

```
//По дате определяет день недели
#include <iostream.h>
void main()
{
    int day,month,year; // день, месяц, год
    int c,y; // столетие и год в столетии
    int m; // месяц по древнеримскому календарю
    int d; // день недели
    cout<<"Input the date:\nday,month,year: \n";
    cin>>day>>month>>year;
    if (month == 1 || month == 2)
        year--; // январь и февраль относятся
        //к предыдущему году
    m = month - 2; // год начинается с марта
    if (m <= 0) m += 12; // для января и февраля
    // здесь m - номер месяца по римскому календарю
    c = year / 100;
    y = year - c*100;
    d = (day+(13*m-1)/5+y+y/4+c/4-2*c+777)%7;
    switch (d)
    {
        case 1: cout<<"Monday"; break;
        case 2: cout<<"Tuesday"; break;
        case 3: cout<<"Wednesday"; break;
        case 4: cout<<"Thirsday"; break;
        case 5: cout<<"Friday"; break;
        case 6: cout<<"Saturday"; break;
        case 0: cout<<"Sunday"; break;
    }
    cout<<endl;
}
```

7. Закрепление изученного. Самостоятельная работа. 12 мин

7.1 Работа с опорным конспектом по образцу. Вычисление кусочно-заданной функции

$$A = \begin{cases} x+1, & x < 0 \\ 2 \cdot x, & 0 \leq x < 10 \\ 0, & x \geq 10 \end{cases}$$

```
#include <iostream.h>
void main()
{double A, x;
cout<<"\n Input x";
cin>>x;
    if (x<0) A=x+1;
        else if (x<10)    A=2*x;
            else A=0;
    cout<<A;
}
```

7.2 Самостоятельная работа по карточкам.

8. Домашнее задание.

2 мин.

[1] стр.237-268, вопросы №№4, 6, 7, 9 (стр. 268-270),

[2] стр.85-93, вопросы №№2, 5, 6, 7 (стр.109-110), упражнение 11 (стр. 111)

самост. “Операторы перехода” ([2] стр.102-106),

[4] №№ 92, 96, 98, 100, 101

Задача 1. Составить программу для решения линейного уравнения $ax = b$.

Задача 2. Квадратный метр площади в построенном доме стоит 270 у.е. (курс Республиканского банка ДНР). Стоимость квартиры С определяется по формуле: $C = SK \cdot \Pi \cdot K$, где SK – стоимость 1 кв. метра площади, Π – площадь квартиры, K – коэффициент, алгоритм определения которого следующий: 1-3 или 13-14 этажи – K=1; 4-7 этажи – K=2,5; 8-9 этажи – K=1,8; 10-12 этажи – K=1,5. Рассчитать стоимость квартиры (в рублях), если известны площадь, этаж и курс у.е.

9. Итог урока.

2мин

Выставление оценок. Что узнали нового?

Задания для теста**1. Какие из следующих утверждений не верны?**

- а) Выражения, соединенные операциями `&&` и `||`, вычисляются слева направо; вычисления прекращаются, как только становится известна истинность или ложность результата.
- б) Приоритет логических операций `&&` и `||` выше, чем у операций сравнения
- в) Использование оператора `break` является обязательным в операторе `switch`.
- г) `case` и `default` могут размещаться в структуре `switch` в произвольном порядке
- д) Выражение, стоящее за ключевым словом `switch` может быть выражением произвольного типа.
- е) Оператор `switch` всегда можно заменить вложенными условными операторами

Ответ: б), в), д).

2. Написать эквивалентное выражение, не содержащее операции !

- а) `!(a > b)`
- б) `!(2*a == b+4)`
- в) `!(a < b && c < d)`
- г) `!(a < 2 || a > 5)`

Ответ: а) `a <= b` б) `2*a > b+4 || 2*a < b+4`

в) `a >= b || c >= d` г) `a >= 2 && a <= 5`

3. Эквивалентны ли следующие фрагменты программы:

- а) `if (e1) if (e2) S1; else S2;`
- б) `if (e1) { if (e2) S1; else S2; }`
- в) `if (e1) { if (e2) S1; } else S2;`

Замечание: `e1` и `e2` - выражения допустимого типа; `S1` и `S2` - произвольные операторы.

Ответ: а) и б)

4. Чему будет равно `s` после следующих действий?

```
int i=25, s;
```

```
if (i = 1) s= i++;  
else s= i--;
```

- а) 25
- б) 2
- в) 31
- г) 26
- д) 24

Ответ: б)

5. Каков результат выражения $(x < 10) ? x = 25 : x++$, если $x = 30$?

- а) 25
- б) 2
- в) 31
- г) 26
- д) 24

Ответ: в)

6. Записать выражение для проверки истинности высказывания:

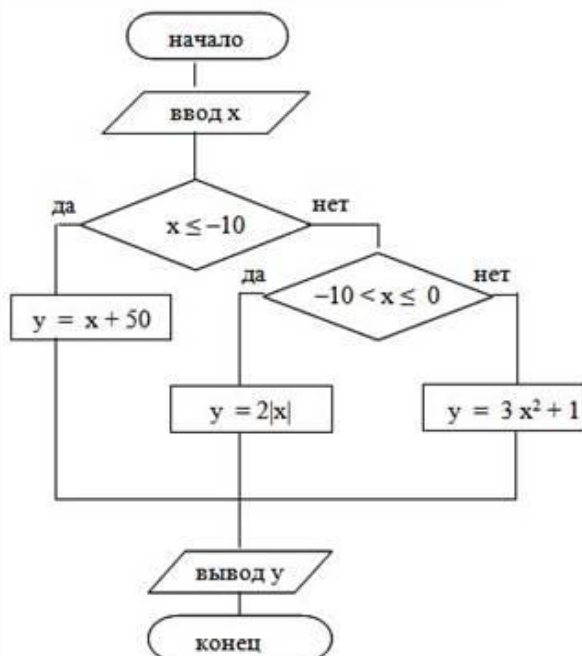
- 1) Все три числа a, b, c положительные
 - 2) Из трёх чисел a, b, c хотя бы одно равно нулю
 - 3) Два числа x, y разных знаков
 - 4) Точка с координатами (x, y) лежит в 1 или 3 координатной четверти
 - 5) Треугольник со сторонами a, b, c равносторонний
- а) $!(a \leq 0 \parallel b \leq 0 \parallel c \leq 0)$
 - б) $x > 0 \ \&\& \ y > 0 \parallel x < 0 \ \&\& \ y < 0$
 - в) $a = 0 \parallel b = 0 \parallel c = 0$
 - г) $a > 0 \parallel b > 0 \parallel c > 0$
 - д) $a * b * c = 0$
 - е) $x * y < 0$
 - ж) $x * y > 0$
 - з) $a = b \ \&\& \ b = c$
 - и) $a > 0 \ \&\& \ b > 0 \ \&\& \ c > 0$

Ответы: 1- и, а; 2- в, д; 3- е; 4- б, ж; 5- з

Карточки с заданиями для самостоятельного решения

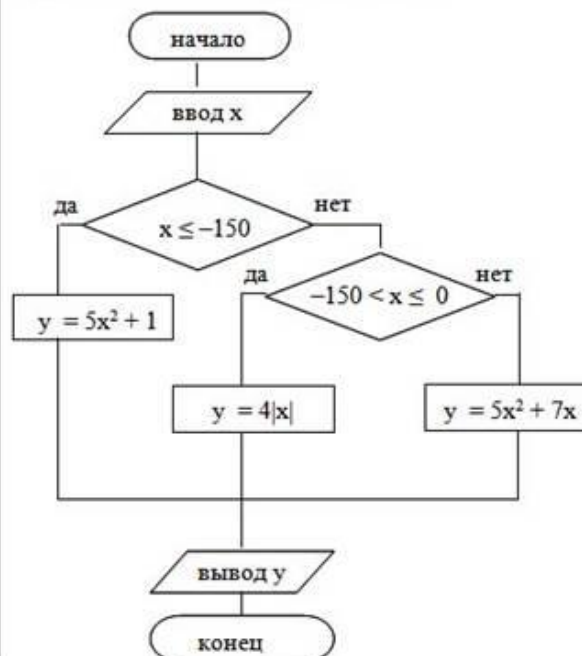
Карточка № 1

По представленной блок-схеме составьте программу вычисления функции y при заданном с клавиатуры значении x .



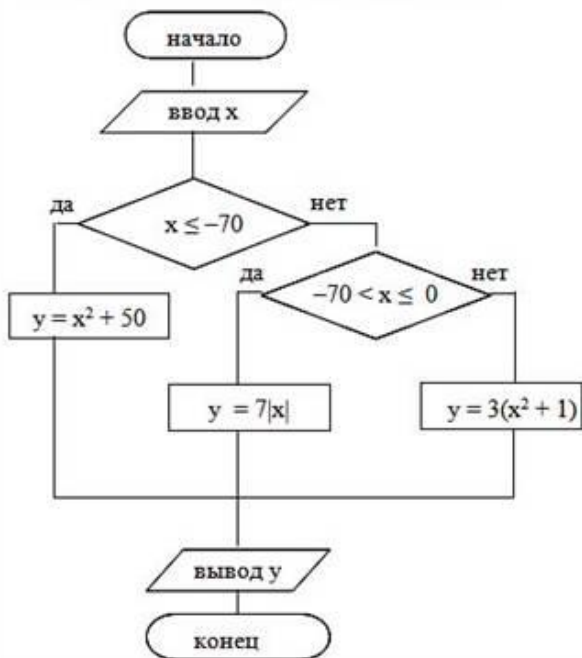
Карточка № 2

По представленной блок-схеме составьте программу вычисления функции y при заданном с клавиатуры значении x .



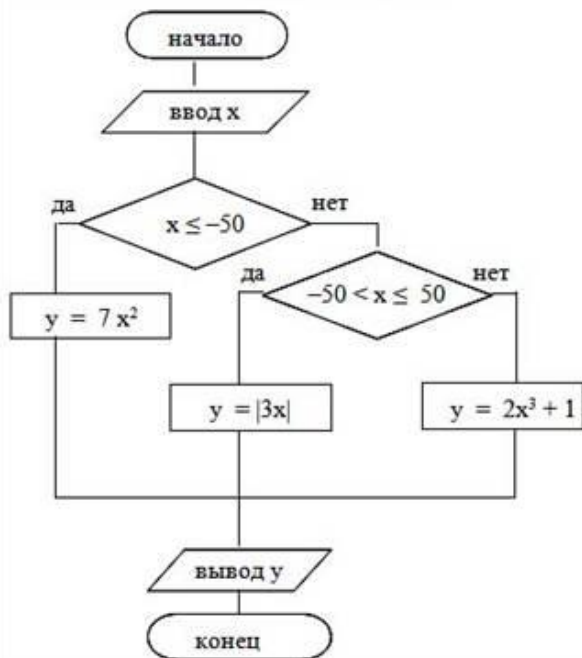
Карточка № 3

По представленной блок-схеме составьте программу вычисления функции y при заданном с клавиатуры значении x .



Карточка № 4

По представленной блок-схеме составьте программу вычисления функции y при заданном с клавиатуры значении x .



Карточка № 1 Составьте блок-схему и программу для вычисления функции y в зависимости от заданного с клавиатуры значения аргумента x: $y = \begin{cases} -x & \text{при } x \leq -1, \\ x^2 & \text{при } -1 < x \leq 0, \\ 1 & \text{при } x > 0. \end{cases}$	Карточка № 2 Составьте блок-схему и программу для вычисления функции y в зависимости от заданного с клавиатуры значения аргумента x: $y = \begin{cases} 1 & \text{при } x \leq -1, \\ x & \text{при } -1 < x \leq 2, \\ \frac{1}{x^2} & \text{при } x > 2. \end{cases}$
Карточка № 3 Составьте блок-схему и программу для вычисления функции y в зависимости от заданного с клавиатуры значения аргумента x: $y = \begin{cases} -3 & \text{при } x \leq -3, \\ x & \text{при } -3 < x \leq 0, \\ -x^2 & \text{при } x > 0. \end{cases}$	Карточка № 4 Составьте блок-схему и программу для вычисления функции y в зависимости от заданного с клавиатуры значения аргумента x: $y = \begin{cases} \frac{1}{x^2} & \text{при } x \leq -1, \\ x^2 & \text{при } -1 < x \leq 1, \\ x & \text{при } x > 1. \end{cases}$

Опорный конспект по теме «Организация ветвлений в программе»

Ветвления в программах на C/C++ можно осуществить с помощью:

- ✓ Условной операции ?:
- ✓ Условного оператор if
- ✓ Оператора множественного выбора switch

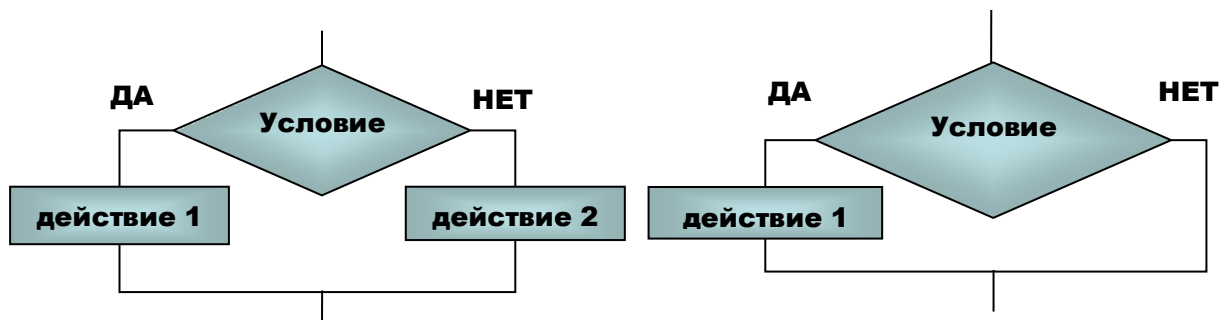
Условная операция

$\text{expr1} ? \text{expr2} : \text{expr3}$

$\text{max} = (\text{a} > \text{b}) ? \text{a} : \text{b}$

Первым вычисляется выражение expr1 . Если оно истинно, то вычисляется expr2 , и оно становится значением условного выражения в целом. Если expr1 ложно, то вычисляется expr3 , и уже оно становится значением условного выражения в целом.

Условный оператор



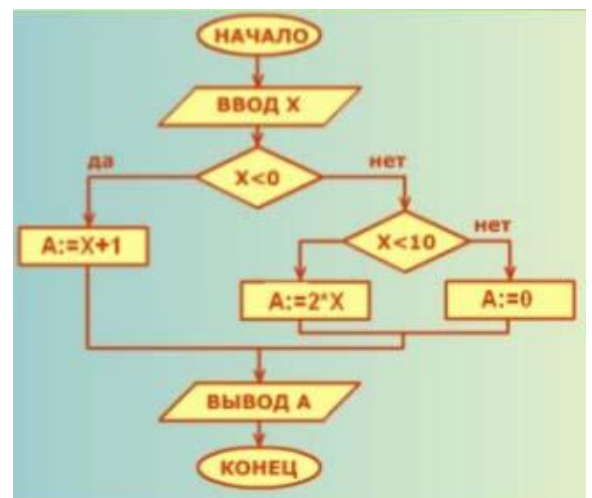
Полное ветвление

Пример Вычислить A по формуле:

$$A = \begin{cases} x+1, & x < 0 \\ 2 \cdot x, & 0 \leq x < 10 \\ 0, & x \geq 10 \end{cases}$$

```
#include <iostream.h>
void main()
{double A, x;
cout<<"\n Input x";
cin>>x;
    if (x<0) A=x+1;
        else if (x<10) A=2*x;
            else A=0;
    cout<<A;
}
```

Неполное ветвление



Оператор выбора switch

```
#include <iostream.h>
void main() {
    char c; int a,b;
    cout<<"Input operation\n";
    cin>>c;
    cout<<"Input a,b\n";
    cin>>a>>b;

    switch (c) {
        case '+': cout<<a+b; break;
        case '-': cout<<a-b; break;
        case '*': cout<<a*b; break;
        case '/': cout<<a/b; break;
        case '%': cout<<a%b;
    break;
        default:
            cout<<"error \n";
    }
}
```

Схема выполнения оператора switch:

- вычисляется выражение-селектор;
- это значение последовательно сравнивается со значениями константных выражений за метками case;
- если значения совпадают, управление передаётся соответствующему помеченному оператору и выполняются операторы и всех последующих ветвей;
- если ни одно значение не совпадает управление передаётся оператору с меткой default, и выполняются операторы и всех последующих ветвей;
- если ни одно значение не совпадает, и нет оператора, помеченного меткой default, управление передаётся оператору, следующему за оператором switch.

Замечание1. Ветви case и default могут располагаться в произвольном порядке

Замечание2. Выражение-селектор и константы выбора могут быть любого перечисляемого типа

Задания для самоподготовки студентов по теме «Организация ветвлений в программах»

1. Оплата за пользование электроэнергией производится по разным тарифам в зависимости от того, установлены ли в доме электрические или газовые плиты. В первом случае (электрические плиты) стоимость 1 кВт/час составляет 2 руб., во втором – 2.5 руб. Составить программу расчета суммы оплаты за пользование электроэнергией за месяц, если известен расход электроэнергии (количество потребленных в течение месяца кВт/час).
2. Оплата за пользование услугами сети Интернет осуществляется по входящему трафику согласно следующей схеме: от 0 до 100 Мбайт – фиксированная плата 120 руб., сверх 100 Мбайт – 2 руб. за каждый Мбайт входящего трафика. Составить программу расчета оплаты за пользование услугами сети Интернет, если известен объем входящего трафика.
3. Кредитное общество выдает денежные кредиты сроком на 1 год. Через год кредит должен быть возвращен вместе с процентами. Величина процентов зависит от суммы кредита: если сумма меньше или равна 100000 руб., начисляется 12% , если больше 100000 руб. – 10%. Составить программу вычисления размера возвращаемой клиентом суммы, если известен размер кредита.
4. Кредитное общество выдает клиентам денежные кредиты на различные сроки. Процент, под который выдается кредит, зависит от срока, на который выдается кредит. Если срок меньше или равен 12 месяцам, начисляется 12% , если больше 12 месяцев – 15%. Составить программу вычисления размера возвращаемой клиентом суммы, если известен размер выданного кредита и срок, на который предоставляется кредит.
5. Фирма представляет автомобили напрокат. Стоимость услуги определяется как произведение оценочной стоимости автомобиля на некоторый коэффициент. Величина коэффициента зависит от длительности проката: если количество дней проката меньше или равно 3, коэффициент равен 0.001, если больше 3 – 0.0005. Составить программу вычисления размера оплаты за прокат автомобиля, если известны оценочная стоимость автомобиля и количество дней проката.
6. Магазин реализует товары, предоставляя покупателям скидки в зависимости от суммы, на которую приобретен товар. Если сумма покупки составляет меньше 10000 руб., скидка не предоставляется. При покупке на сумму от 10000 руб. до 30000 руб. скидка составляет 5%. При покупке на сумму свыше 30000 руб. размер скидки – 10% от суммы покупки. Составить программу вычисления оплачиваемой покупателем суммы, если известна сумма покупки без скидки.

7. Размер оплаты за посещение ребенком детского сада зависит от среднего месячного дохода на 1 человека в семье, который определяется делением совокупного дохода семьи на количество человек в семье. Если при этом доход на 1 человека меньше 5000 руб. в месяц, предоставляется скидка, равная 50% от базовой оплаты. Если доход на 1 человека больше либо равен 5000 руб. и меньше либо равен 7000 руб., предоставляется скидка 30%, если доход больше 7000 руб., скидка не предоставляется. Составить программу определения оплаты при заданных значениях совокупного месячного дохода семьи, количественного состава семьи и базовой ставки оплаты за пользование детским садом.
8. Стоимость туристической путевки определяется в зависимости от времени года. Базовая стоимость путевки умножается на сезонный коэффициент, который определяется следующим образом: октябрь, май, январь – коэффициент 1; июнь, июль август, сентябрь – коэффициент 1.2; ноябрь, декабрь, февраль, март, апрель – коэффициент 0.8. Составить программу определения стоимости путевки, если известна ее базовая стоимость и месяц, на который приобретается путевка.
9. Транспортное предприятие доставляет заказчикам грузы. Стоимость услуги определяется дальностью поездки: если дальность меньше 10 км, то тариф составляет 20 руб./км, от 10 км до 100 км – 15 руб./км, свыше 100 км – 10 руб./км. Составить программу расчета стоимости доставки груза, если известна дальность поездки.
10. Транспортное предприятие перевозит грузы из пункта А в пункт В, расстояние между которыми составляет 100 км. Стоимость доставки определяется весом перевозимого груза. Базовый тариф равен 20 руб./км и применяется для грузов весом от 500 до 1000 кг. Если вес груза меньше 500 кг, базовый тариф умножается на коэффициент 0.8, если груз больше 1000 кг, применяется коэффициент 1.5. Составить программу определения стоимости доставки при заданном весе перевозимого груза.
11. Страховая компания занимается страхованием домашнего имущества. При заключении договора клиент оплачивает страховой взнос, размер которого определяется как произведение объявленной стоимости страхуемого имущества (страховая сумма) на тариф. Величина тарифа определяется величиной страховой суммы: если страховая сумма меньше либо равна 100000 руб., тариф равен 0.01, если страховая сумма больше 100000 руб., тариф равен 0.02. Составить программу определения размера страхового взноса, если известна страховая сумма.
12. Медицинская клиника проводит платный прием пациентов. За каждый прием пациент оплачивает фиксированную сумму, определенный процент от которой составляет заработная плата врача, осуществляющего прием. Указанный процент зависит от категории врача: для врача высшей категории – 40%, первой категории – 30%, второй категории – 20%. Составить программу расчета размера

отчисления на заработную плату врача, если известна стоимость приема и категория врача.

- 13.Авиакомпания предоставляет скидки на билеты в определенные периоды года (январь, февраль, июнь, июль, август). Если полет выполняется в январе или феврале, скидка составляет 40% от базовой стоимости билета, в летнее время скидка равна 20%. Составить программу определения стоимости билета, если известен базовый тариф и время выполнения полета (номер месяца).
- 14.Сельхозпредприятие реализует продукцию оптовым покупателям. Существует базовая цена реализации на каждый вид продукции. В зависимости от количества реализуемой продукции покупателям могут предоставляться скидки: если вес продукции больше 100 кг, предоставляется скидка 5%, если вес больше 300 кг, скидка составляет 10% от базовой цены. Составить программу определения стоимости реализуемой продукции, если известна базовая цена и количество продукции.
- 15.Фирма сдает в аренду клиентам офисные помещения. Существует базовая цена 1 кв. м. сдаваемых помещений. Если общая площадь арендуемых клиентом помещений меньше 20 кв. м., базовая цена увеличивается на 10%, если же площадь больше 100 кв.м., базовая цена уменьшается на 10%. Если площадь от 20 до 100 кв.м., применяется базовая цена. Составить программу вычисления стоимости аренды офисных помещений клиентом фирмы, если известна базовая цена 1 кв. м. и общая площадь арендуемых клиентом помещений.